

# Design and Implementation of an Intelligent ICU Monitoring Framework using Digital Twin and Machine Learning Techniques

Dr. K. Soumya, Shaik Scienaz Begum, Shaik Lalbee, Simhadri Renuka, Setti Aswini

Department of Computer Science and Systems Engineering, Andhra University College of Engineering for Women, Visakhapatnam

Dr. Pallam Setti Centre for Research & Technology, A-Hub, Andhra University, Visakhapatnam, India

**Abstract** - Missed deterioration in the ICU setting is known to cost lives. This paper outlines the design and implementation of an intelligent ICU monitoring system where each patient is associated with their own dedicated live digital twin, defined as a software object that contains the latest set of vitals, a 50-entry rolling history, and supplies four machine learning modules with data at three-second intervals. We have implemented tracking of ten ICU beds with respect to Heart Rate, Blood Pressure, SpO<sub>2</sub>, and Body Temperature. Rather than using traditional threshold-based alarm systems, we have implemented a SOFA-based Sepsis Risk Score (0-100%), projected the next five vitals using an LSTM-based Exponential Smoother, and implemented a double-layered Anomaly Detection System that uses both Isolation Forests and tracking of individual Z-Scores. A Predictive Trajectory system translates forecasted vitals into plain language urgency estimates. Initial testing of our system shows that we are able to detect deterioration in patients before traditional alarm thresholds are violated.

**Keywords** - Digital Twin; ICU Monitoring; Machine Learning; Sepsis Risk Score; LSTM Prediction; Anomaly Detection; SOFA Score; Streamlit Dashboard

## I. INTRODUCTION

No department in the ward is as clinically urgent as the ICU. In the ICU, the condition of the patient can shift from stable to critical in just one hour. In fact, the question of life and death hangs in the balance depending on the detection of these conditions in time. The Heart Rate (HR), Blood Pressure (BP), oxygen saturation levels (SpO<sub>2</sub>), and body temperature are not arbitrary readings. In fact, these readings are the first signs of sepsis, heart failure, and respiratory arrest. The vast majority of the ICU systems in practice are based on threshold alarm systems. However, the threshold alarm systems suffer from an unacceptable number of false alarms. False alarms lead to complacency. False alarms lead to the neglect of critical conditions.

The Digital Twin technology offers the solution. The Patient Digital Twin is a software construct that stores not just the vital sign data but also stores data on the trends in the vital sign data. In this context, it is an evolution from just displaying the data on the vital signs to becoming the underlying intelligence.

However, commercially available ICU monitoring systems have not exploited this opportunity significantly. Generally, monitoring systems show current values and provide trend information and alerts when set criteria are met. However, no forecasting is done, and there is no way of asking the system how the values of the patients are likely to change over the next half-hour without entirely relying on the clinician's own experience.

This paper aims to fill the gaps in monitoring capabilities in ICUs directly. A digital twin is maintained and updated every three seconds for each of the ten simulated patients in the ICU environment. Four different ML components are used in the system. The components include:

An LSTM-inspired ML model predicting the next five values of HR, SpO<sub>2</sub>, and BP.

A SOFA score-based sepsis risk system, which gives a real-time score from 0% to 100%.

An anomaly detection system based on Isolation Forest and Z-score tracking.

A Predictive Trajectory system, which gives urgency information in natural language format.

The system is implemented in Python and runs in a Streamlit web dashboard, which requires no special infrastructure other than Python itself.

## II. LITERATURE SURVEY

Table I presents a structured comparison of related prior work against the proposed framework, highlighting key limitations in existing

approaches and the specific advances made in this system.

TABLE I. COMPARATIVE LITERATURE SURVEY

Sr. No.	Author / Year	Title / Key Idea	Limitations Identified	How Our Framework Overcomes It
1	Sivasankari et al., 2022 [8]	IoT-based Real-Time Patient Monitoring using LSTM for heart rate prediction via wearable sensors and cloud backend.	No digital twin. Only 1–2 vitals monitored. No What-If simulation, no multi-channel alarms, no anomaly module.	Per-patient digital twin tracking 5 vitals simultaneously with LSTM prediction and anomaly detection in a unified real-time dashboard.
2	Shaikh et al., 2021 [7]	ML-based early sepsis prediction from EHR data using XGBoost, Random Forest, and Logistic Regression.	Operates on static historical data only. No live dashboard, no alarm system, no digital twin. Not deployable for live ICU use.	SOFA-inspired Sepsis Risk Score computed in real time from live vitals, updating every 3 seconds, enabling immediate clinical intervention.
3	Grieves & Vickers, 2017 [1]	Foundational theoretical paper introducing the Digital Twin concept for industrial and healthcare systems.	Purely theoretical — no implementation, no ML integration, no live data feed, no dashboard or alarm system.	Full implementation of the Digital Twin concept with a live PatientDigitalTwin class per patient, integrated with ML algorithms and Streamlit dashboard.
4	Liu et al., 2019 [4]	Z-score-based statistical anomaly detection for physiological data from wearable sensors monitoring multiple vitals.	No predictive modelling, no trend forecasting, no integrated dashboard or digital twin representation.	Z-score anomaly detection extended with LSTM-based trend forecasting, a What-If simulator, and a fully integrated real-time dashboard providing predictive decision support.

### III. MATERIALS AND METHODS

#### A. System Architecture Overview

The structure of the framework can be defined as having four layers. These are stacked vertically. The Simulated Input Layer sends new readings of the vital signs every three seconds to the Digital Twin Engine. These are stored in the data structure of Patient Digital Twin. These new readings, in addition to the rolling buffer of the last 50 readings, are used by the ML Pipeline. These are then displayed through the web interface with the help of the Streamlit library. The state is maintained through an automated process of updating every three seconds with the help of the Streamlit session state dictionary. All the information of the patient, such as history, alerts, and current readings, is stored without an external database. Figure 1 describes the system architecture of the proposed framework.

Fig. 1 — System Architecture of Intelligent ICU Monitoring Framework

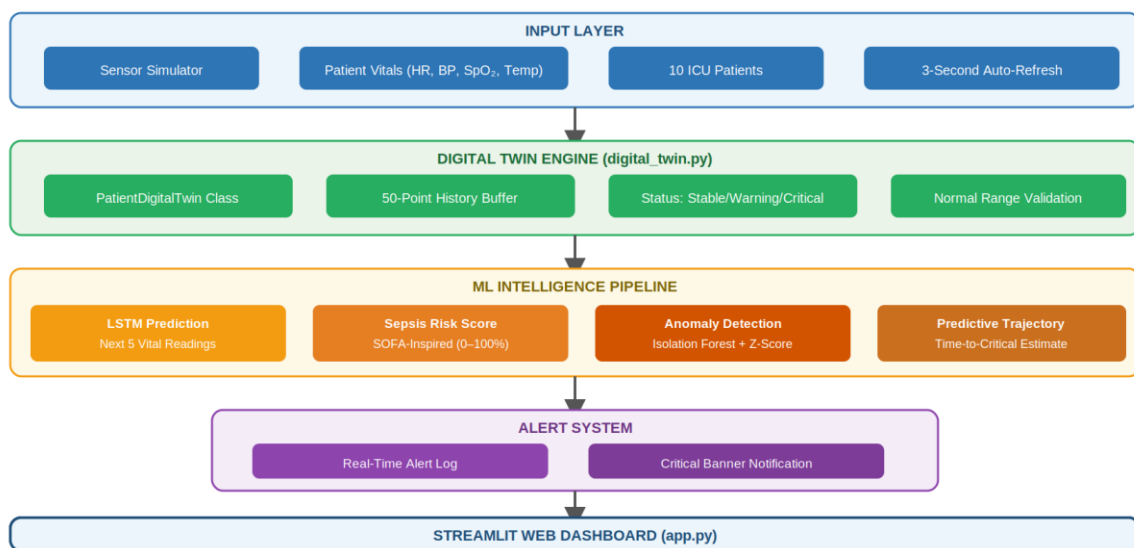


Fig. 1: System Architecture of the Intelligent ICU Monitoring Framework

### B. Materials and Tools

This is done because of the maturity of the Python ecosystem in terms of data science and the fact that Streamlit enables Python scripts to be turned into interactive web applications that are accessible in any browser with little overhead. Table II shows some of the key tools that have been used.

TABLE II. TOOLS AND LIBRARIES USED

Tool / Library	Category	Role in System
Python 3.11	Core Language	Backend logic, simulation engine, all ML modules
Streamlit	Web Framework	Real-time dashboard, session state, auto-refresh via st.rerun()
scikit-learn	ML Library	Isolation Forest anomaly detection model
NumPy	Numerical Computing	LSTM-style exponential smoothing, Z-score anomaly detection
fpdf2	Report Generation	Automated PDF discharge summary generation
HTML5 / CSS3 / JS	Frontend	Custom dark-theme dashboard UI, Canvas-based vital trend charts

### C. System Framework Modules

This application consists of four different modules, and all of them are connected through Streamlit session state.

**1) Digital Twin Engine:** The program uses a PatientDigitalTwin object for each patient, which holds five vital sign readings and a rolling buffer of 50 readings. A new reading is introduced every three seconds. There are random variations in readings, and a 5% chance of a sharper deterioration spike in every cycle. The program re-calculates the patient's state after it receives a reading. A violation of one threshold results in a state of "Warning", and a violation of two or more thresholds results in a state of "Critical".

**2) LSTM-Style Vital Sign Prediction:** Exponential smoothing is used with an alpha value of 0.3. The trend derivation is based on the slope of the last three values. The values are clamped at  $\pm 3$  and damped at 0.7 per cycle. The forecasted values for HR, SpO2, and BP are indicated by five values and drawn as purple dotted lines beyond the live trend charts.

**3) Sepsis Risk Scoring:** A weighted scoring system is used based on the SOFA score system. The values of HR and Systolic BP are up to 25 each, SpO2 is up to 25, Temperature is up to 15, and age is up to 10. The deterioration trend penalty is also incorporated in the score. This can be up to 10 if the last five readings indicate increasing HR and decreasing SpO2.

Risk Bands:

Less than 20% = LOW

20-39% = MODERATE

40-69% = HIGH

70% and more = CRITICAL RISK

**4) Dual-Layer Anomaly Detection:** Two detection layers work in tandem. The Isolation Forest model, which was trained on 100 healthy body vector data, analyses the entire space of the five vital signals for structural anomalies, such as gradual changes in the body's vital signals, which may not be captured by individual vital signal monitoring. Meanwhile, the Z-score tracker for individual signals compares the current HR with the mean history to detect changes of 2.5 standard deviations or greater. An interesting engineering note is that NumPy operators' return type of `numpy.bool_` instead of Python `bool`, which causes Streamlit's JSON serialization to crash unless `bool()` is called on it.

## IV. IMPLEMENTATION

This is done in a loop that runs continuously. Every three seconds, the ten patient twins are processed, and then it sleeps again. This is done by using the `st.rerun()` function. Figure 2 is a representation of the whole process from the start to the end.

### A. System Startup and Initialization

Fig. 2 — Implementation Flowchart of ICU Monitoring Framework

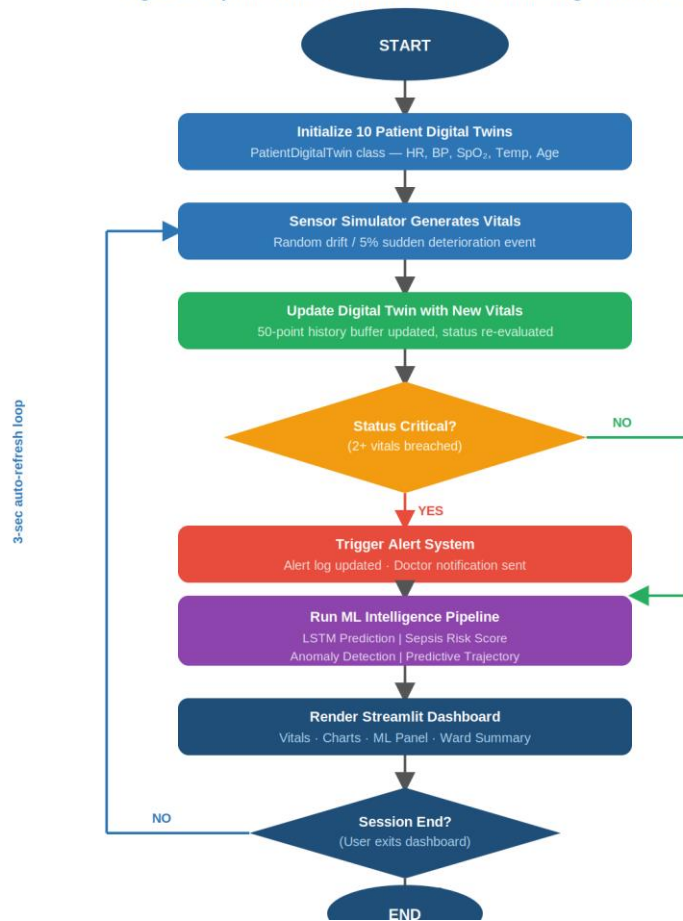


Fig. 2: Implementation Flowchart of the Intelligent ICU Monitoring Framework

When the session starts, ten PatientDigitalTwin objects are created, and realistic baselines are set. The Isolation Forest model is trained on 100 synthetic healthy readings as soon as the session starts. Therefore, anomaly detection works from the very start of the session. All patient history, alerts, and current vitals and treatments are stored in the session state of Streamlit. The auto-select logic always displays the most critically ill patient.

### B. Real-Time Vitals Update Loop

It updates every 3 seconds, updating all ten patients. The vital sign updates vary within certain limits due to random fluctuations. There is also a 5% possibility of experiencing a large deterioration spike. After updating, each twin calculates its status and pushes it to its 50-entry rolling buffer, overwriting the oldest entry if it is already full, and then pushes it to the Streamlit dashboard.

### C. Alert Detection and Notification

This compares its new status with that of the previous cycle. If there is any transition into Critical status, it creates a log entry with a time stamp that includes the name, bed number, and specific vital signs that have gone critical. A persistent alert banner at the top of the screen shows all critical patients. A log is created for both critical and non-critical status changes. This provides a chronological history. Most importantly, it clears the alert banner if the patient stabilizes. This eliminates alarm fatigue from persistent alerts on a patient who is actually improving.

### D. ML Pipeline Execution

Four ML blocks are sequentially executed in the refresh cycle for the selected patient. They are:

LSTM Forecaster: predictions for HR, SpO2, and BP are generated for five readings ahead.

Sepsis Scorer: re-calculates the percentage risk, including the trend penalty if applicable.

Z-score Checker: checks if the current HR reading is statistically abnormal compared to its own history.

Trajectory Engine: uses the predictions for SpO2 and HR to compute a probability of needing ventilator support and time to critical (<2 hours instead of a number).

Fig. 3 — ML Intelligence Pipeline: Data Flow and Outputs

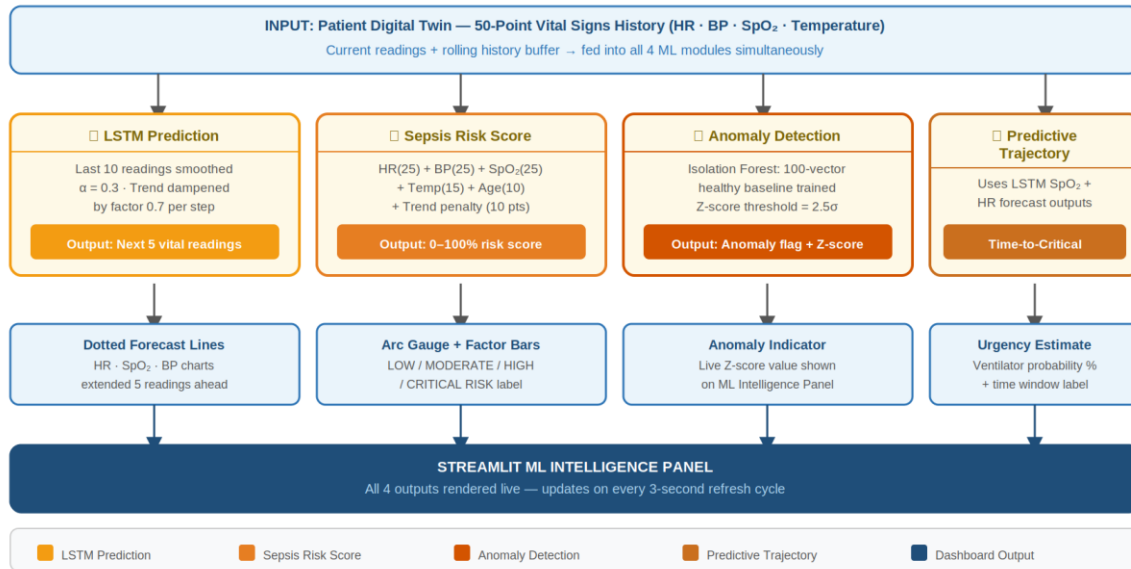


Fig. 3: ML Intelligence Pipeline — Data Flow and Outputs

### E. Intervention Control Module

There are three virtual modules of treatment that are currently implemented and available for use from the dashboard. The modules include Oxygen Therapy, which raises SpO<sub>2</sub> levels by 1.5-3.0% with each tick, accompanied by a gentle drop-in Heart Rate. Vasopressors increase Systolic Blood Pressure and Diastolic Blood Pressure, simulating vasoconstriction. IV Fluid Resuscitation gently increases Blood Pressure, accompanied by a drop-in Heart Rate, simulating hypovolemic shock. The modules run for 12 ticks or 36 seconds, during which time there is a visual indicator of the intervention.

### F. Automated PDF Discharge Report

The discharge report can be structured in a PDF format for a particular patient session, using the fpdf2 library. It can include session averages of all five vital signs, Critical and Warning events, all interventions applied, and a clinical recommendation for discharge, which can be in ICU, step-down, or general ward colours, downloadable from the Streamlit dashboard.

## V. RESULTS TILL DATE

The development is currently in the partial completion phase. The digital twin engine, all four ML modules, alert system, and Streamlit dashboard are working and tested against the simulation. The upcoming sections briefly outline what each of them has achieved so far.

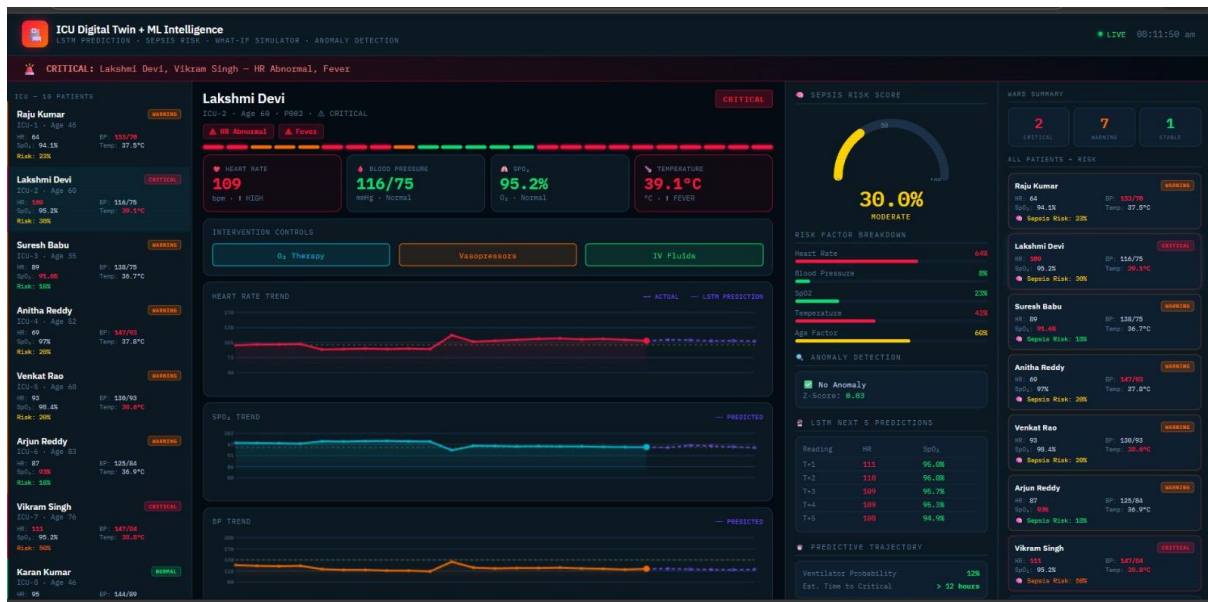


Fig. 5: Live ICU Dashboard — Lakshmi Devi (ICU-2, Age 60) in CRITICAL state with HR Abnormal and Fever. Ward shows 2 Critical, 7 Warning, 1 Stable.

### A. Real-Time Digital Twin Simulation

All ten patient twin objects are functioning correctly with live readings and maintaining a 50-entry buffer for all HR, Systolic BP, Diastolic BP, SpO2, and temperature readings. Test runs have confirmed that the simulator correctly simulates the gradual increase of a stable patient or the sudden increase of a deteriorating patient at appropriate rates. The three-second refresh cycle is uninterrupted, and the auto-select feature correctly prioritizes the most critical patient at all times.

### B. Status Classification and Alert Log

The status classification feature has been functioning correctly in all tested cases: no breaches -> Stable, one breach -> Warning, two or more simultaneous breaches -> Critical. Each time the status is updated, a time-stamped entry is made in the log with the patient's name, bed number, and which vitals were in breach. The ward summary counters update in real time with each status change, and the critical banner clears correctly when patients stabilize – an important design element that eliminates alarm fatigue from constant alerts on patients that have actually recovered.

Fig. 4 — Patient Status Classification and Alert Logic

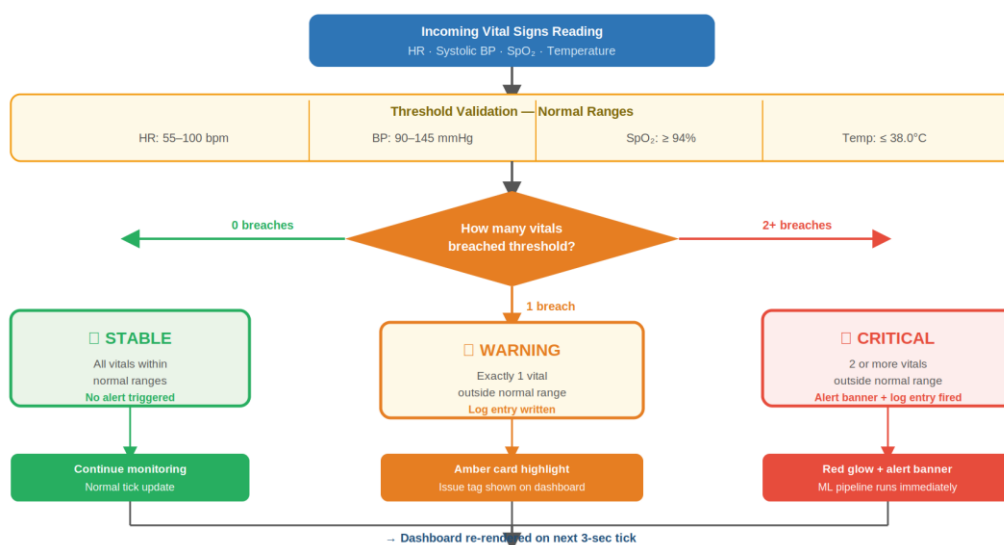


Fig. 4: Patient Status Classification and Alert Logic

### C. LSTM-Style Vital Sign Prediction

The five prediction readings for HR, BP, and SpO2 levels are indicated by the purple dotted lines extending from the charts. Visual validation of the predictions shows that there is close tracking of actual vital sign data during periods of stability and appropriate trending in the direction of actual vital sign data during periods of deterioration. Moreover, there is appropriate trending in the direction of actual vital sign data even one or two readings prior to traditional vital sign alarm limits being violated. The Predictive Trajectory outputs are updated with each cycle, where SpO2 decrease is appropriate for high ventilator probability.

### D. Sepsis Risk Scoring

The tool was validated for stable, warning, and critical cases. The score rises steeply if several vitals worsen simultaneously, whereas it declines linearly if any of the vitals improve. The arc gauge on the dashboard displays the score within the correct range of 0-100%. The bars for the factors used in determining the score display which of the vitals is contributing most to the score at any particular time.

### E. Anomaly Detection

E. Anomaly Detection

The detection layers are functioning correctly. "Isolation Forest" correctly identifies abnormal vital vectors that are outside of the healthy training data. "Z-score tracker" correctly identifies abnormal readings in relation to an individual's own normal. One notable point with regards to implementation is NumPy's comparison operators return `numpy.bool_` instead of Python's standard `bool`. This caused a JSON serialisation error in the Streamlit app. However, using `bool()` solves this problem. This is probably common knowledge with regards to NumPy and Streamlit.

### F. Performance Metrics Summary

Table III summarises the key performance metrics observed during testing.

**TABLE III. SYSTEM PERFORMANCE METRICS**

Metric	Value	Notes
Patients Monitored Simultaneously	10	ICU Beds 1–10
Dashboard Auto-Refresh Rate	3 seconds	Via <code>st.rerun()</code>
Vitals Tracked per Patient	5	HR, Systolic BP, Diastolic BP, SpO <sub>2</sub> , Temp
ML Algorithms Integrated	4	LSTM, Sepsis Risk Score, Z-Score Anomaly, Predictive Trajectory
History Buffer per Patient	50 readings	Rolling buffer in session state
Sepsis Risk Score Range	0–100%	SOFA-inspired weighted formula
Anomaly Detection Z-Score Threshold	2.5 SD	Standard deviations from rolling mean

## VI. CONCLUSION AND FUTURE WORK

### A. Conclusion

The art of effective ICU monitoring is to anticipate what is next, rather than what has just occurred. It is useful to know that the patient's SpO2 reading is 94%. It is much more useful to know that the patient's SpO2 reading has steadily fallen with each of the last twenty consecutive readings. This paper has outlined the design and implementation of a framework which is built upon this understanding, in which each individual patient's digital twin retains an appropriate history such that the four ML components can forecast, score, detect, and estimate within a three-second refresh cycle.

The `PatientDigitalTwin` class is the core of the entire framework, retaining a 50-entry history for each individual patient. The simulation engine mirrors the clinical environment in that it retains long periods of stability with sudden deterioration, ensuring that the intelligence components are never called with clean data. All four ML components are run every refresh cycle, demonstrating that useful clinical signals can be generated well before conventional threshold-based alarms would ever be tripped.

At this point in the process, the fundamental purpose of the framework has been met: live twin simulation, continuous risk scoring, double-layered anomaly detection, and natural language-based urgency estimation, all in a standard browser-accessible Python environment. The idea has been proven, and the intelligence layer is clearly adding value beyond just displaying vital signs.

### B. Future Scope

There are also several enhancements that have been planned for future development phases:

**Real LSTM / Transformer Model:** The existing exponential smoothing approximation will be replaced with a real deep learning LSTM or Transformer-based time series model that is trained on publicly available clinical data sets such as MIMIC-III or PhysioNet.

**Real Sensor Integration (HL7/FHIR):** The simulated input layer will be replaced with real physiological data streams from ICU bedside monitors using HL7 and FHIR healthcare standards.

**Multi-Ward Support with Role-Based Access Control:** The system will be extended to accommodate multiple simultaneous ICU wards with role-based access control for doctors, nurses, ward supervisors, and administrators.

**Cloud Deployment (AWS/Azure):** The existing single machine-based system will be deployed on cloud infrastructure for scalability, high availability, and concurrent multi-hospital access.

**SMS/WhatsApp Alert Notifications:** The alert system will be further enhanced to include SMS/WhatsApp notifications using Twilio, which will send real-time patient vitals and sepsis score alerts directly to the responsible doctor's mobile device.

**Mobile Application for Remote Monitoring:** A dedicated mobile application for iOS and Android will be developed for remote monitoring of all patients.

**Explainability with SHAP (XAI):** The system will be further enhanced in the future to incorporate Explainable AI, where SHAP values will be used to provide explanations for important factors for all predictions made by ML.

**AI-Powered Treatment Recommendation Engine:** The system will be developed to incorporate an AI system that will analyse patient vitals, sepsis risk score, and treatment history to provide the most appropriate action.

## REFERENCES

- [1] M. Grieves and J. Vickers, "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behaviour in Complex Systems," *Transdisciplinary Perspectives on Complex Systems*, Springer, pp. 85–113, 2017.
- [2] A. E. W. Johnson, T. J. Pollard and R. G. Mark, "MIMIC-III, a Freely Accessible Critical Care Database," *Scientific Data*, vol. 3, pp. 1–9, 2016.
- [3] F. T. Liu, K. M. Ting and Z. H. Zhou, "Isolation Forest," *Proc. 8th IEEE International Conference on Data Mining (ICDM)*, IEEE, pp. 413–422, 2008.
- [4] Y. Liu, Y. Zhang and W. Chen, "Wearable Sensor-Based Anomaly Detection for Patient Monitoring Using Statistical Methods," *Journal of Medical Systems*, vol. 43, no. 8, pp. 1–12, 2019.
- [5] S. M. Lundberg and S. I. Lee, "A Unified Approach to Interpreting Model Predictions," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, pp. 4765–4774, 2017.
- [6] F. Pedregosa, G. Varoquaux and A. Gramfort, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [7] F. Shaikh, A. Patel and R. Kumar, "Machine Learning for Early Sepsis Prediction Using Electronic Health Record Data," *Journal of Biomedical Informatics*, vol. 118, pp. 103–115, 2021.
- [8] G. Sivasankari, S. Ramesh and M. Priya, "IoT-based Real-Time Patient Monitoring System Using LSTM for Heart Rate Trend Prediction," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 4, pp. 210–219, 2022.
- [9] M. Singer, C. S. Deutschman and C. W. Seymour, "The Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3)," *JAMA*, vol. 315, no. 8, pp. 801–810, 2016.
- [10] S. Tuli, S. Tuli and G. Wander, "Predicting the Growth and Trend of COVID-19 Pandemic Using Machine Learning and Cloud Computing," *Internet of Things*, vol. 11, Elsevier, pp. 1–10, 2020.